

*A.F. 01W*

MS158551.01/MSFTP199US

CERTIFICATE OF MAILING

I hereby certify that this correspondence (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313 - 1450.

Date: 9-27-04

*Himanshu S. Amin*  
Himanshu S. Amin

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re patent application of:

Applicant(s): Dennis Angeline, *et al.*

Examiner: Farhood Moslehi

Serial No: 09/820,433

Art Unit: 2152

Filing Date: March 29, 2001

Title: SYSTEM AND METHOD FOR BRIDGING OBJECT SYSTEMS

**Mail Stop Appeal Brief – Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450**

---

**APPEAL BRIEF**

---

Dear Sir:

Applicants submit this brief in triplicate in connection with an appeal of the above-identified patent application. A credit card payment form is filed concurrently herewith in connection with all fees due regarding this document. In the event any additional fees may be due and/or are not covered by the credit card, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1063 [MSFTP199US].

**I. Real Party in Interest (37 C.F.R. §41.37(c)(1)(i))**

The real party in interest in the present appeal is Microsoft Corporation, the assignee of the present application.

**II. Related Appeals and Interferences (37 C.F.R. §41.37(c)(1)(ii))**

Appellants, appellants' legal representative, and/or the assignee of the present application are not aware of any appeals or interferences which will directly affect, or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**III. Status of Claims (37 C.F.R. §41.37(c)(1)(iii))**

Claims 1-42 are pending in the application. The rejection of claims 1-42 is being appealed.

**IV. Status of Amendments (37 C.F.R. §41.37(c)(1)(iv))**

No claim amendments have been entered after the Final Office Action.

**V. Summary of Invention (37 C.F.R. §41.37(c)(1)(v))**

The subject invention relates to bridging communications between disparate object systems. A service (*See* page 8, lines 12-18, Fig. 1 ref. 32) is provided to enable bi-directional communications between object systems (*See* page 7, lines 9-19, Fig. 1 refs. 20, 22) that may support different languages, architectures and object system interfaces. The service generates a wrapper and associated interfaces (*See* page 7, lines 20-28, Fig. 1 refs. 30, 34) the object systems utilize and enables the respective object systems to be insulated from implementation details and inconsistencies of the other object systems. Thus, code associated with one object system may transparently interact with code written for another object system *via* the wrappers and associated interfaces. In order to bridge communications in accordance with the present invention, a plurality of system inconsistencies and/or disparities are considered and provided for within the wrappers. The subject invention also includes a system for bridging objects. The system includes means for activating an interface wrapper (*See* page 8, lines 12-18, Fig. 1 ref. 32) from a first object system according to interface implementations of a second object

system. This includes means for utilizing the interface wrapper (*See* page 8, lines 26-28, Fig. 1 refs. 34, 42) to facilitate transparent communications between managed and unmanaged object systems, means for directing communications (*See* page 10, lines 7-9, Fig. 2 ref. 54) between the object systems, and means for proxying (*See* page 10, lines 20-23, Fig. 2 refs. 50, 60) the respective object systems in order to marshal data between the object systems.

**VI. Grounds of Rejections (37 C.F.R. §41.37(c)(1)(vi))**

A. Whether claims 1-17, 22, 25-33 and 37-42 are unpatentable under 35 U.S.C. §103(a) over Jordan (US 6,016,392) in view of Schaefer *et. al.* (US 6,629,192).

B. Whether claims 18-21, 23, 24 and 34-36 are unpatentable under 35 U.S.C. §103(a) over Jordan (US 6,016,392) in view of Schaefer *et. al.* (US 6,629,192) and in further view of Foody *et. al.* (US 5,732,270).

**VII. Argument (37 C.F.R. §41.37(c)(1)(vii))**

A. **Rejection of Claims 1-17, 22, 25-23, and 37-42 Under 35 U.S.C. §103(a)**

Claims 1-17, 22, 25-33 and 37-42 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Jordan (US 6,016,392) in view of Schaefer *et. al.* (US 6,629,192). Withdrawal of this rejection is respectfully requested for at least the following reasons.

- i. ***Jordan and Schaefer et. al. alone or in combination fail to teach or suggest all the limitations set forth in independent claims 1, 26, 37, 40 and 42 and the claims that depend there from.***

To reject claims in an application under §103, an examiner must establish a *prima facie* case of obviousness. A *prima facie* case of obviousness is established by a showing of three basic criteria. First, there must be some suggestion or motivation, either in the

references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) ***must teach or suggest all the claim limitations***. See MPEP §706.02(j). The ***teaching or suggestion to make the claimed combination*** and the reasonable expectation of success ***must be found in the prior art and not based on the Applicant's disclosure***. See *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991) (emphasis added).

Independent claim 1 (and similarly independent claims 26, 37, 40 and 42) recite ***communications between managed and unmanaged object systems***. Managed and unmanaged object systems relate to software environments that execute objects under markedly different circumstances depending on whether or not a respective object operates according to managed architectural operating conditions or unmanaged architectural operating conditions. For instance, ***managed and unmanaged object systems*** are distinguished in one aspect by the type of object lifetime management supported in such systems. This management of objects takes place in volatile memory regions (*e.g.*, Ram, cache memory) along with the storage and execution of the objects as well. In an unmanaged system, techniques such as reference counting are employed to manage object lifetime such as how is the object memory reclaimed when the object is no longer in use. These type systems require the object to perform the reference counting.

For managed systems, techniques such as garbage collecting are employed whereby a component such as a garbage collector reclaims object memory outside of normal object processes. Thus, memory reclamation is highly dependent on the object to perform such services in an unmanaged systems and mostly independent of the object in a managed system. Until the advent of the subject invention, objects in one type of object system could only communicate to objects operating in the ***same*** type of object system (*e.g.*, managed objects only communicating with other managed objects, and unmanaged objects only communicating with other unmanaged objects). Interface wrappers support communications between these disparate type object systems in a unique and efficient manner.

In sharp contrast to the claimed invention, Jordan neither discloses nor suggests ***communications between managed and unmanaged object systems***. Rather, Jordan merely discloses communications between objects in the ***same*** type of object system such as a Component Object model (COM) system. With respect to COM systems, these would represent well-known object models operating according to the rules of reference counting employed by unmanaged object systems (*See* Fig. 2A ref. 2A50, and col. 4 line 56 of Jordan). Notably, Jordan does not disclose or remotely suggest managed object systems let alone teach or suggest how objects in a managed environment may communicate with objects in an unmanaged environment as recited in the claimed invention. Jordan merely teaches an alternative interface technique for objects operating in unmanaged operating systems - not bridging communications between objects operating in managed and unmanaged object systems as claimed.

Schaefer *et al.* does not make up for the aforementioned deficiencies of Jordan with respect to facilitating communications between managed and unmanaged object systems. In fact, Schaefer *et al.* is entirely ***unrelated*** in any manner with respect to the claimed invention. Schaefer *et al.* discloses a system whereby non-volatile memory is broken into “unmanaged space” and “managed space.” These terms have ***no contextual*** relationship to the claimed invention for a number of reasons. For instance, managed and unmanaged space in Schaefer *et al.* refers to segmenting non-volatile memory such as ROM which is entirely different than a segmented volatile memory whereby objects execute, and the managed side objects are processed by a garbage collector and the unmanaged side objects are reference counted.

On the other hand, objects in applicants’ claimed invention execute in volatile memory such as RAM which is entirely infeasible with respect to the non-volatile memory disclosed in Schaefer *et al.* Moreover, the term “managed” in the context of Schaefer *et al.* merely refers to a portion of a flash device which is managed by a non-volatile ***storage*** manager which controls access to a portion of a “BIOS” which is entirely different than facilitating communications between objects in managed and unmanaged ***object*** systems. Therefore, the meaning of the terms managed and unmanaged in Schaefer *et al.* are completely ***unrelated*** to the terms managed and unmanaged recited in the respective claims.

Since neither Jordan nor Schaefer *et al.* alone or in combination teach or suggest communications between managed and unmanaged object systems as is readily apparent from the foregoing discussion illustrating that the cited references fail to teach or suggest the limitations of independent claims 1, 26, 37, 40, and 42 and claims that depend therefrom - reversal of this rejection is respectfully requested.

**ii. *Jordan and Schaefer et al. when combined fail to provide teaching or suggestion in either reference leading to the claimed combination.***

In addition to the reasons cited above, there is clearly no motivation provided in the references themselves to combine the cited references as suggested by the Examiner since Schaefer *et al.* is entirely unrelated to object systems of the claimed invention that by definition employ objects that execute substantially out of volatile memory. Also, the memory manager system of Schaefer *et al.* does not remotely apply to managed and unmanaged **object** systems. Moreover, there is no reason disclosed or suggested from within the references (or from without) to combine Schaefer *et al.* with Jordan since Jordan relates to object interactions between **all** objects operating in a COM environment and Schaefer *et al.* is unrelated to both object execution and to corresponding interactions between such objects.

The mere fact that the reference can be modified does not render the modification obvious unless the referenced art also suggests the desirability of the modification. *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990). Furthermore, a teaching or suggestion to make the claimed combination and a reasonable expectation of success must both be found in the prior art, not in applicants' disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

In this case, there can be no expectation of success even if the references were combined since Schaefer *et al.* does not bear any resemblance to the object execution environment of the claimed invention. Since neither Jordan nor Schaefer *et al.* alone or

in combination teach or suggest the invention as claimed, it is respectfully submitted that this rejection be reversed.

**B. Rejection of Claims 18-21, 23, 24, and 34-36 Under 35 U.S.C. §103(a)**

Claims 18-21, 23, 24 and 34-36 stand rejected under 35 U.S.C. §103(a) over Jordan (US 6,016,392) in view of Schaefer *et. al.* (US 6,629,192) and in further view of Foody *et. al.* (US 5,732,270). Withdrawal of this rejection is respectfully requested for at least the following reasons.

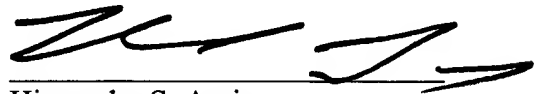
- i. Jordan and Schaefer et. al. in further view of Foody et. al. alone or in combination fail to teach or suggest all the limitations set forth in independent claims 1, 26, 37, 40 and 42 and the claims 18-21, 23, 24 and 34-36 that depend there from.*

This rejection should be reversed and withdrawn for at least the following reasons. Foody *et al.* does not make up for the aforementioned deficiencies of Jordan or Schaefer *et al.* alone or in combination with respect to independent claims 1, 26, 37, 40 and 42 described above. Notably, Foody *et al.* does not teach or suggest employment of **interface wrappers** as recited in these claims. Rather, Foody *et al.* teaches creating a **redundant** proxy object that **mirrors** a foreign object. Thus, any manipulations to the proxy are mirrored in the foreign object. One clear disadvantage to creating a redundant object is this type of communication consumes more memory than the claimed invention that employs interface wrappers to communicate to a single object in a disparate object system. Moreover, Foody *et al.* does not disclose or suggest communications **between managed** and **unmanaged** object systems as recited in the subject claims. Furthermore, there does not appear to be any motivation or suggestion in the cited references themselves to combine these references as suggested by the Examiner. In view of the foregoing, it is readily apparent that the cited references do not make obvious applicants' invention as recited in the subject claims; and it is respectfully requested that this rejection be reversed.

**VIII. Conclusion**

For at least the above reasons, the claims currently under consideration are believed to be patentable over the cited references. Accordingly, it is respectfully requested that the rejections of claims 1-42 be reversed.

Respectfully submitted,  
AMIN & TUROCY, LLP



Himanshu S. Amin  
Reg. No. 40,894

AMIN & TUROCY, LLP  
24<sup>th</sup> Floor, National City Center  
1900 East 9<sup>th</sup> Street  
Telephone: (216) 696-8730  
Facsimile: (216) 696-8731



**IX. Appendix of Claims (37 C.F.R. §41.37(c)(1)(viii))**

- 1) A system for bridging disparate object systems, comprising:
  - a memory boundary between a managed and an unmanaged object system;
  - a first interface wrapper to bridge communications across the memory boundary between a first object system and a second object system; and
  - a second interface wrapper to bridge communications across the memory boundary between the second object system and the first object system, wherein the first interface wrapper insulates the first object system from interface implementations in the second object system and the second interface wrapper insulates the second object system from interface implementations in the first object system to facilitate transparent communications between the first and second object systems.
- 2) The system of claim 1, wherein the first object system is at least one of a managed object system and an unmanaged object system, and the second object system is at least one of a managed object system and an unmanaged object system.
- 3) The system of claim 1, wherein the first wrapper and second wrapper serve as a proxy to the respective object systems that point to a stub within the wrappers in order to marshal data between the object systems.
- 4) The system of claim 1, wherein the first wrapper queries for type information from the second object system and forms interfaces from methods exposed from the type information.
- 5) The system of claim 1, wherein the second wrapper calls the first object system and determines an interface by casting to a type and examining an exception.
- 6) The system of claim 5, wherein an adapter object is provided to map interfaces of an unknown type in the first object system to a known type in the second object system.

- 7) The system of claim 1, wherein the first object system is reference counted and the second object system is traced.
- 8) The system of claim 7, wherein the first wrapper maintains a traced reference for the second object system and reference counts interfaces utilized by the first object system.
- 9) The system of claim 7, wherein the second wrapper holds a traced reference for the second object system and releases interfaces utilized by the first object system.
- 10) The system of claim 7, further comprising a garbage collector to reclaim objects within the second object system, wherein unmanaged objects are reclaimed based upon the reference count in the first object system.
- 11) The system of claim 1, wherein object identities are maintained by utilizing a single managed wrapper per each object.
- 12) The system of claim 11, wherein a specialized wrapper is defined that subtypes off of a generic wrapper to simulate a class.
- 13) The system of claim 1, further comprising a bridging services component to detect an unmanaged interface call and direct a managed client to an unmanaged object.
- 14) The system of claim 13, wherein the unmanaged interface call is detected through a vtable reference from the second object system.
- 15) The system of claim 1, wherein one or more objects belonging to the first and second object systems are activated *via* at least one of an early bound and late bound manner.

- 16) The system of claim 15, wherein a late bound interface is employed to provide late bound activation.
- 17) The system of claim 15, wherein early binding is provided at compile time *via* type information derived from a foreign object system.
- 18) The system of claim 17, wherein type information is provided from at least one of a type library export and type library import tool.
- 19) The system of claim 1, wherein the first object system utilizes results returned on a method call and the second object system utilizes exceptions.
- 20) The system of claim 19, wherein results are mapped to exceptions and exceptions are mapped to results.
- 21) The system of claim 1, wherein object reusability is provided *via* an inner object and outer object relationship.
- 22) The system of claim 1, wherein intra object communications is provided *via* wrappers.
- 23) The system of claim 22, wherein inter object communications is provided *via* proxies within the wrappers.
- 24) The system of claim 1, wherein calls are routed to a foreign object system according to environment partitioning rules of the foreign object system.
- 25) A computer-readable medium having computer-executable components for executing the system of claim 1.

- 26) A method for bridging objects, comprising:  
activating an interface wrapper from a first object system according to interface implementations of a second object system; and  
utilizing the interface wrapper to facilitate transparent communications between managed and unmanaged object systems.
- 27) The method of claim 26, further comprising,  
providing bridging services to direct the communications between the object systems.
- 28) The method of claim 26, further comprising,  
proxying the respective object systems from a stub within the wrappers in order to marshal data between the object systems.
- 29) The method of claim 26, further comprising,  
querying type information from the second object system; and  
forming interfaces from methods exposed from the type information.
- 30) The method of claim 26, further comprising,  
determining an interface by casting to a type; and  
examining an exception resulting from the caste.
- 31) The method of claim 26, further comprising,  
maintaining object identities by utilizing a single managed wrapper per each object.
- 32) The method of claim 31, further comprising,  
creating a specialized wrapper that subtypes off of a generic wrapper to simulate a class.

- 33) The method of claim 26, further comprising,  
activating objects *via* at least one of an early binding and a late binding.
- 34) The method of claim 26, further comprising,  
providing type information from at least one of a type library export and type library import tool.
- 35) The method of claim 26, further comprising,  
mapping results to exceptions; and  
mapping exceptions to results.
- 36) The method of claim 26, further comprising,  
routing calls to a foreign object system according to environment partitioning rules of the foreign object system.
- 37) A system for bridging objects, comprising:  
means for activating an interface wrapper from a first object system according to interface implementations of a second object system; and  
means for utilizing the interface wrapper to facilitate transparent communications between managed and unmanaged object systems.
- 38) The system of claim 37, further comprising,  
means for directing communications between the object systems.
- 39) The system of claim 37, further comprising,  
means for proxying the respective object systems in order to marshal data between the object systems.

- 40) A signal facilitating object communications, comprising:
- a signal for communicating between managed and unmanaged object systems;
  - an interface wrapper activated *via* the signal from a first object system according to interface implementations of a second object system, wherein the interface wrapper facilitates transparent communications between the one or more object systems.
- 41) The signal of claim 40, wherein the signal is communicated over at least one of a network system and a wireless system.
- 42) An object system bridge, comprising:
- at least one interface wrapper; and
  - a bridge service to enable the interface wrapper and facilitate transparent communications between at least one of a managed object system and an unmanaged object system;
- wherein the interface wrapper insulates the at least one of the managed object system and the unmanaged object system from interface implementations in at least one other managed object system and unmanaged object system.

**X. Evidence Appendix (37 C.F.R. §41.37(c)(1)(ix))**

None.

**XI. Related Proceedings Appendix (37 C.F.R. §41.37(c)(1)(x))**

None.